IN THE CLAIMS

Please cancel Claims 23-27 without prejudice, and amend Claim 18 as follows:

1.-11. [Cancelled]

12. (Previously presented) An integrated circuit, fabricated using the method comprising:

creating a customized description language model of an integrated circuit design by:

editing a first file specific to said design;

defining the location of at least one library file;

generating a script using said first file, said library file, and user input information; and

running said script to create said customized description language model;

generating a netlist which is descriptive of the circuitry of said integrated circuit;

compiling said netlist and said hardware description model to produce a compiled integrated circuit design;

fabricating at least one mask representing said compiled integrated circuit design; and

fabricating said integrated circuit using said at least one mask;

wherein said act of creating is performed at a high level of abstraction.

13. (Previously presented) The integrated circuit of Claim 12, wherein the act of editing comprises selecting at least one of a plurality of input parameters associated with said design, said at least one parameter being selected from the group comprising:

(i) custom instruction sets;

(ii) cache configurations;

(iii) memory interface configurations; and

(iv) system architecture configurations.

14. (Previously presented) The integrated circuit of Claim 12, wherein the act of generating a netlist comprises generating a list of logic devices and their interconnections.

15. (Previously presented) The integrated circuit of Claim 12, wherein the act of fabricating said integrated circuit comprises defining physical features on a semi-conductive substrate via a lithographic process.

16. (Previously presented) The integrated circuit of Claim 12, further comprising synthesizing said design based on said description language model.

17. (Previously presented) The integrated circuit of Claim 13, wherein the act of editing is performed interactively with the user using a display.

18. (Presently amended) An apparatus adapted to generate integrated circuit designs, comprising;

a processor capable of running a computer program;

a storage device operatively coupled to said processor, said storage device being capable of storing at least a portion of a computer program;

an input device, operatively coupled to said processor, capable of receiving input from a user and transmitting said input to said processor; and

a computer program resident at least in part on said storage device, said computer program adapted to receive said input relating to a constrained set of design variables relating to a basecase processor configuration from said user and perform the following acts based on said input:

editing a first file specific to said integrated circuit design;

defining the location of at least one library file;

generating a script using said first file, said library file, and user input information; and

running said script to create said description language model of said integrated circuit design.

19. (Previously presented) The apparatus of Claim 18, wherein said description language model is a hardware description language (HDL).

20. (Previously presented) The apparatus of Claim 18, wherein said computer program is further adapted to perform the acts comprising:

generating a second file based on said description language model for use with a simulation; and

simulating said design using said second file.

21. (Previously presented) The apparatus of Claim 20, wherein said computer program is further adapted to perform the act comprising running synthesis scripts based on said description language model in order to synthesize said integrated circuit design.

22. (Previously presented) The apparatus of Claim 18, wherein said processor comprises a digital microprocessor, and said storage device comprises magnetic media.

23.- 39. [Cancelled]

40. (Previously presented) A system for generating integrated circuit designs at a high level of abstraction, comprising:

a processor;

a storage device in data communication with said processor, said storage device being capable of storing and retrieving a computer program; and

a computer program stored within said storage device and adapted to run on said processor, said computer program comprising;

a user-configurable macro-instruction having at least a first user-selectable element, said first-selectable element being selected from the group comprising;

(i) a plurality of custom instructions;

(ii) a plurality of cache configurations;

(iii) a plurality of memory interface configurations; and

(iv) a plurality of system architecture configurations;

a first algorithm capable of generating a script based on selections made by a user from said at least first user selectable element; and

a second algorithm capable of running said script to generate a description language model of an integrated circuit design

41. (Previously presented) The system of Claim 40, wherein said computer program further comprises a second user-selectable element, said second user-selectable element allowing said user to select one of a plurality of process technology options.

42. (Previously presented) The system of Claim 40, wherein said first user-selectable element is selected by the act of reading a pre-configured data file.

43.- 46. [Cancelled]

47. (Previously presented) A method of generating the design of an integrated circuit rendered in a hardware description language, said method being performed at a high level of abstraction and comprising the acts of:

selecting a process technology;

editing a first file specific to the design, said act of editing comprising selecting at least one user-configurable parameter selected from the group comprising;

(i) processor instructions;

(ii) cache configuration;

(iii) memory interface configuration; and

(iv) system architecture configuration;

defining the location of at least one library file;

generating a script using said first file and said library;

running said script to create a customized hardware description language model of the design; and

running a synthesis algorithm to synthesize a file descriptive of said design.

48. (Previously presented) A system for generating integrated circuit designs at a high level of abstraction, comprising:

means for processing digital data;

means for data storage in data communication with said processor means, said means for data storage being capable of storing and retrieving a computer program; and

a computer program stored within said means for data storage and adapted to run on said processor means, said computer program comprising;

means for selecting a process technology;

a user-configurable macro-instruction having at least one user-selectable element, said user-selectable element being selected from the group comprising;

i) a plurality of instructions;

(ii) a plurality of cache configurations;

(iii) a plurality of memory interface configurations; and

(iv) a plurality of system architecture configurations;

means for generating a script based on said user selectable element and said process technology; and

means for running said script to generate a description language model of an integrated circuit design.

49.-59. [Cancelled]

60. (Previously presented) A method of designing a configurable processor, the method comprising:

generating, at a high level of abstraction, a processor specification having a user-definable portion, the user-definable portion of said specification including at least one user-defined instruction having a function associated therewith; and

based on said processor specification, generating a description of a hardware implementation of said configurable processor.

61. (Previously presented) The method of Claim 60, wherein said act of generating a description comprises generating a description including control logic necessary for the execution of said at least one user-defined instruction.

62. (Previously presented) The method of Claim 61, wherein said act of generating a description of a hardware implementation comprises describing at least an instruction execution pipeline having a plurality of stages, said control logic including portions associated with said stages.

63. (Previously presented) The method of Claim 60, wherein said act of generating a description comprises generating a description having at least one element selected from the group consisting of:

(i) registers; (ii) condition code choices; and (iii) scratchpad RAM.

64. (Previously presented) The method of Claim 60, wherein said act of generating a description comprises generating a description having at least one library of multimedia extensions.

65. (Previously presented) The method of Claim 60, further comprising simulating said configurable processor using at least said description.

66. (Previously presented) The method of Claim 65, wherein said act of simulating comprises:

running at least one script to generate simulation data;

running at least one simulation using at least said simulation data; and

determining the adequacy of said configurable processor based at least in part on said act of running.

67. (Previously presented) The method of Claim 60, further comprising synthesizing said configurable processor using at least said description.

68. (Previously presented) The method of Claim 67, wherein said act of synthesizing comprises:

running at least one synthesis script to generate synthesis data;

and evaluating the adequacy of said synthesis data based at least in part on at least one design criterion.

69. (Previously presented) The method of Claim 68, wherein said at least one design criterion comprises:

at least one specific processor performance criterion; and

at least one processor die size criterion.

70. (Previously presented) The method of Claim 68, further comprising:

revising at least one design element when said act of evaluating indicates that said synthesis data is not adequate;

re-running said at least one synthesis script using said at least one revised design element to generate revised synthesis data;

and re-evaluating the adequacy of said revised synthesis data based at least in part on said at least one design criterion.

71. (Previously presented) The method of Claim 70, wherein said at least one design criterion comprises at least one processor die size criterion, and said act of revising comprises revising at least one library.

72. (Previously presented) The method of Claim 71, wherein said at least one design criterion comprises at least one processor die size criterion, and said act of revising further comprises revising at least one control file.

73. (Previously presented) The method of Claim 70, wherein said at least one design criterion comprises processor clock speed, and said act of revising comprises revising at least one library.

74. (Previously presented) The method of Claim 70, wherein said at least one design criterion comprises processor power consumption, and said act of revising comprises revising at least one netlist (net load).

75. (Previously presented) A method of generating the design of an integrated circuit at a high level of abstraction using a description language, comprising the acts of:

providing an existing processor core configuration;

editing a first file specific to the design, said editing comprising selecting a constrained set of input parameters associated with said configuration, said parameters comprising:

(i) at least one custom instruction;

(ii) a cache configuration; and

(iii) a memory interface configuration;

providing at least one library file;

generating a script using said first file, said library file, and user input information;

running said script to create a customized description language model; and

synthesizing said design based on said description language model.

76. (Previously presented)A method of generating an integrated circuit design at a high level of abstraction, comprising:

providing a user with a plurality of optional instructions, including the ability to generate a customized instruction;

selecting at least one of said plurality of optional instructions;

selecting at least one cache configuration;

defining at least one memory interface;

generating a script based on said at least one optional instruction, cache configuration, and memory interface; and

running said script to generate a hardware description language model of said integrated circuit design.

77. (Previously presented) A description language model of an integrated circuit design generated at a high level of abstraction using the method comprising:

editing a first file specific to said integrated circuit design including selecting a plurality of input parameters associated with said design, said parameters comprising:

(i) at least one extension instruction; and

(ii) a cache configuration;

defining the location of at least one library file;

generating a script using said first file, said library file, and user input information; and

running said script to create said description language model of said integrated circuit design;

wherein said method is performed at a high level of abstraction.

78. (Previously presented) A method of generating an extended processor design at a high level of abstraction, comprising:

providing the user with a basecase processor core configuration having a base instruction set;

providing a user with a plurality of optional instructions adaptable for use with said basecase core;

selecting at least one of said plurality of optional instructions;

selecting at least one cache configuration;

generating a script based on said at least one optional instruction, cache configuration, and basecase core; and

running said script to generate a hardware description language model of said processor design;

wherein said plurality of optional instructions and cache configurations are constrained so as to ensure the functionality of said processor design irrespective of the user's selections.